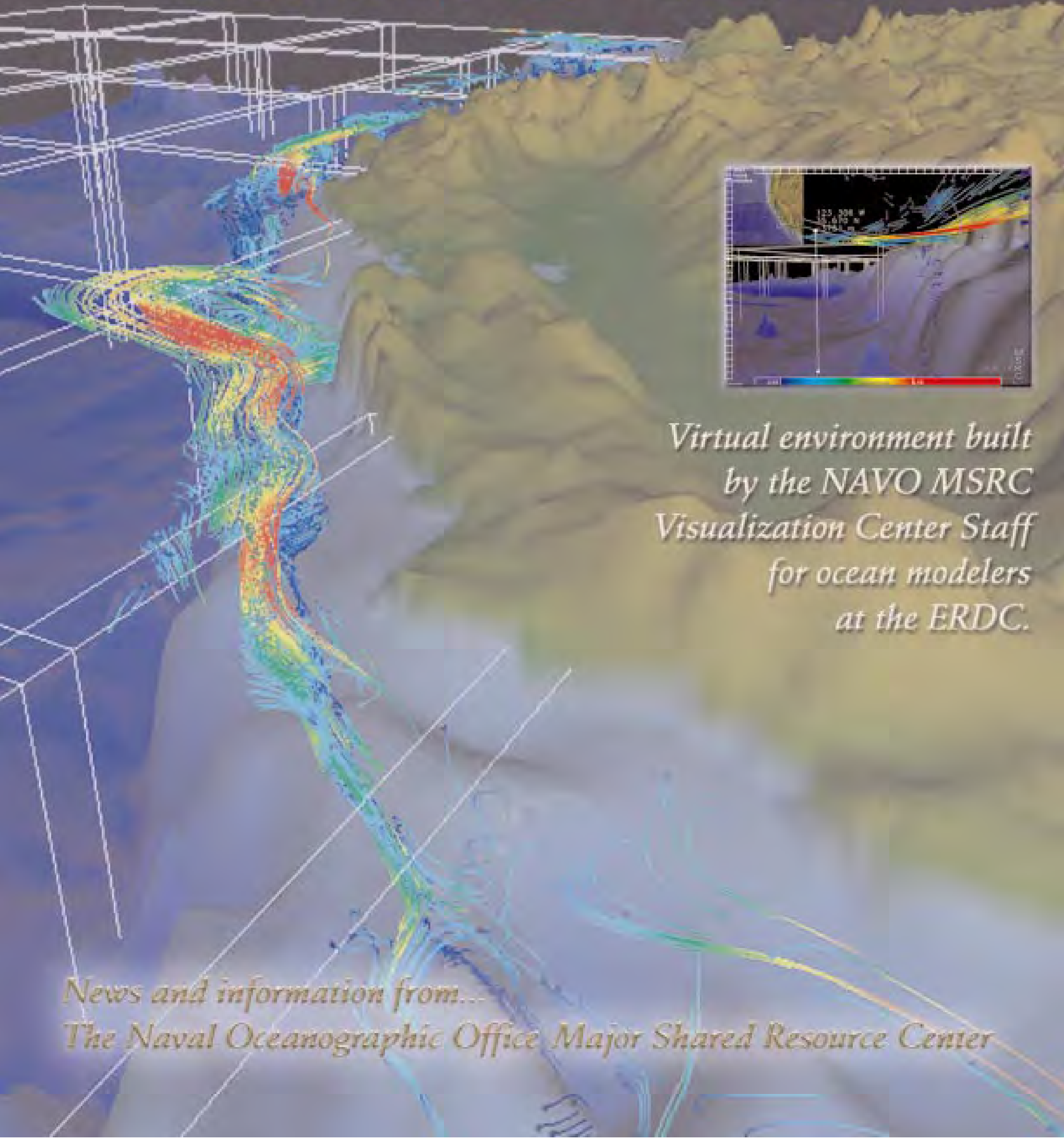
Navigator

NAVO MSRC Spring 2001



Virtual environment built by the NAVO MSRC Visualization Center Staff for ocean modelers at the ERDC.

News and information from...
The Naval Oceanographic Office Major Shared Resource Center

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2001**	2. REPORT TYPE	3. DATES COVERED **00-00-2001 to 00-00-2001**
4. TITLE AND SUBTITLE **NAVO MSRC Navigator. Spring 2001**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Oceanographic Office (NAVO),Major Shared Resource Center (MSRC),1002 Balch Boulevard,Stennis Space Center,MS,39522**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**24**	

The Director's Corner

Steve Adamec, NAVO MSRC Director

Looking Forward and Back at the NAVO MSRC

This year marks the tenth anniversary of the establishment of the Primary Oceanographic Prediction System (POPS) supercomputer center here at the Naval Oceanographic Office. The POPS center, the forerunner of the present NAVO MSRC, initially offered its user community a CRAY Y-MP/8 system with 2.7 gigaflops of peak computing capability. Significantly, it was established to simultaneously serve both research and development (R&D) and operational high performance computing (HPC) requirements within the Navy. Over the past ten years, the center has increased its computing capacity 1000-fold while continuing to serve both the Department of Defense (DoD) R&D and Navy operational HPC needs. The focus on combined R&D and operational HPC processing within one center has yielded significant benefits to DoD, including high systems availability, resilient networking and storage infrastructure and has dramatically improved scheduling of the largest HPC applications. These applications include those associated with the DoD High Performance Computing Modernization Program (HPCMP) challenge projects and the time-critical, global-scale HPC applications for the operational Navy community, which must run multiple times every day of the year.

As we cruise into summer, the preparations for UGC 2001 are almost complete. This year's conference promises to be a good one, affording us the opportunity to extend some Gulf Coast hospitality to a large contingent of the DoD user community. The Shared Resource Combined Advisory Panel (SRCAP) has done an outstanding job of organizing this year's event, and we are privileged to assist them in bringing it to fruition. We look forward to seeing old and new friends in Biloxi this coming June.

Finally, we bid farewell to Mr. Terry Blanchard, who retired in March 2001 after more than thirty years of distinguished Federal service. Terry's contributions to the DoD HPCMP as both Deputy Director and Director of the NAVO MSRC were numerous and substantial, enabling this MSRC to establish, sustain, and enhance a premiere HPC capability for the DoD user community.

About the Cover:

Virtual environment built by the NAVO MSRC Visualization Center staff for ocean modelers at the ERDC. This application allows the researcher to analyze a model output generated from a multiblock grid.

The Naval Oceanographic Office (NAVO) Major Shared Resource Center (MSRC): Delivering Science to the Warfighter

The NAVO MSRC provides Department of Defense (DoD) scientists and engineers with high performance computing (HPC) resources, including leading edge computational systems, large-scale data storage and archiving, scientific visualization resources and training, and expertise in specific computational technology areas (CTAs). These CTAs include Computational Fluid Dynamics (CFD), Climate/Weather/Ocean Modeling and Simulation (CWO), Environmental Quality Modeling and Simulation (EQM), Computational Electromagnetics and Acoustics (CEA), and Signal/Image Processing (SIP).

NAVO MSRC
Code N7
1002 Balch Boulevard
Stennis Space Center, MS 39522
1-800-993-7677 or
help@navo.hpc.mil

NAVO MSRC Navigator
www.navo.hpc.mil/navigator

NAVO MSRC Navigator is a bi-annual technical publication designed to inform users of the news, events, people, accomplishments, and activities of the Center. For a free subscription or to make address changes, contact NAVO MSRC at the address above.

EDITOR:
Gioia Furness Petro, petrogio@navo.hpc.mi

DESIGNERS:
Patti Geistfeld, pgeist@navo.hpc.mil
Kerry Townson, ktownson@navo.hpc.mil
Lynn Yott, lynn@navo.hpc.mil

Contents

Scalable Flow Simulations with Rotating Components

Roger Briley, Mechanical Engineering Department
Lafayette K. Taylor and David L. Whitfield, Aerospace Engineering Department,
Computational Simulation and Design Center, ERC, Mississippi State University
http://www.erc.msstate.edu/simcenter

Scalable parallel computing is greatly advancing the complexity of problems for which analysis and design, based on large-scale complex flow simulations, is becoming feasible. The Computational Simulation and Design Center (SimCenter) at Mississippi State University's Engineering Research Center (ERC) has developed scalable flow simulation software for both multiblock structured grids that have arbitrary block connectivity and for multielement unstructured grids. These flow solvers are capable of high-resolution simulations for very large Reynolds numbers (i.e., $\sim 10^9$).

Two current Office of Naval Research- (ONR-) sponsored Department of Defense (DoD) Challenge projects with allocations at the Naval Oceanographic Office (NAVO) MSRC and United States (U.S.) Army Engineer Research and Development Center (ERDC) Major Shared Resource Center (MSRC) are using these codes to study unsteady viscous flow phenomena associated with underwater vehicles and surface ships.

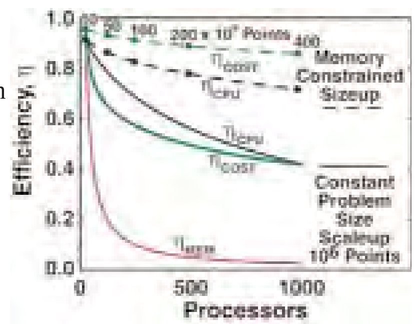
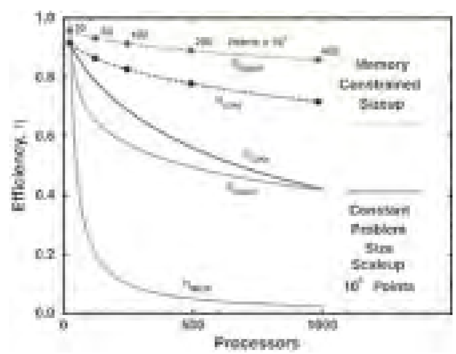


Figure 1a. (Left) *1b.* (below) Scalability Properties from a Semiempirical Performance Model



One project, led by Dr. L. Patrick Purtell, ONR, focuses on submerged wakes in littoral regions and continues a previous Challenge project at the Arctic Region Super Computer Center (ARSC) on submarine maneuvering. The second project, led by Dr. Ki-Han Kim, ONR, concerns surface ship maneuvering and sea keeping. Additionally, the National Aeronautics and Space Administration (NASA) Ames sponsors simulations of tilt-rotor aircraft flows and maneuvering in a study that is related to these projects through a cooperative agreement with the Navy.

All these projects require scalable parallel supercomputing to address the requirements of large-scale unsteady viscous-flow simulations, past complex geometries with dynamic and rotating components.

The parallel algorithms now being used[1-2] have evolved over the past ten years from previous serial algorithms for the unsteady Reynolds-averaged Navier-Stokes equations. They combine multiple-iteration implicit schemes, characteristic-based finite-volume spatial approximations, and numerical flux linearizations with Block-Jacobi Gauss-Seidel relaxation for the innermost iteration to provide scalable concurrency. The semiempirical performance model cited in reference 1 is used to illustrate some of the scalability properties of the structured-grid flow solver for very large-scale problems, and some examples of complex unsteady flow simulations with rotating components are given from recent work of SimCenter researchers.

Semiempirical Model for CPU, Memory, and Cost Efficiencies

A semiempirical performance model has been developed[1] to study the scalability of the parallel solution algorithms as actually implemented for existing and hypothetical computing platforms using Message Passing Interface (MPI).

The parallel algorithms, spatial domain decomposition, and message-passing software framework were specifically designed to provide scalability for complex flow simulations on modern distributed memory architectures. The codes have operated efficiently on T3E, Origin2/3K, Sun Enterprise, SP2/3, and Unix/Linux

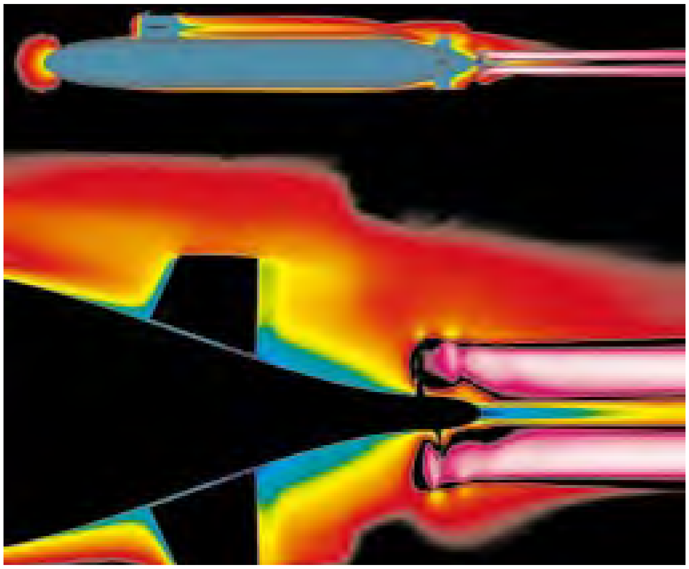


Figure 2. Propelled Notional Submarine in Straight-Ahead Motion

clusters. The actual Central Processing Unit (CPU) times and communications overhead are routinely measured, and observed efficiencies have been consistent with this model for cases run on numerous machines, currently up to 11 million points and 100 processors.

For present purposes, the parallel CPU efficiency $\eta_{cpu} = T_{cpu}/T_{runtime}$ is defined to be the ratio of the total time spent in CPU operations to the total run time, including message passing communications. By definition, the communications overhead is $1-\eta_{cpu}$. The memory efficiency $\eta_{Mem} = Mb_{Used}/Mb_{Reserved}$ is the ratio of processor memory utilized during execution to the total memory reserved.

The total memory includes idle processor memory not used during execution, adjusted for any shared memory actually allocated to other users. An unused memory overhead could be defined as $1-\eta_{Mem}$. Finally, by assuming that hardware costs are apportioned as 50% CPU, 30% memory, and 20% supporting hardware, a cost efficiency for hardware resource utilization can be defined as $\eta_{Cost} = 50\%\eta_{CPU} + 30\%\,\eta_{Mem} + 20\%$. Although hardware costs obviously vary, these assumed percentages at least approximate current market pricing, and other reasonable estimates would not significantly alter the predicted trends.

The performance model[1] estimates CPU and message-passing times based on architectural parameters for each specific machine, including a measured effective CPU mega floating-point operations per second (Mflop) rate (as compiled), a rate for loading and unloading of message buffer arrays, MPI software bandwidth and latency, and the number of processors. Solution algorithm parameters are also used, including number of grid points and subiteration cycles, floating-point operation count, and number and average length of messages. Although the rotating interface conditions are implemented in a scalable form, the performance model does not yet include messages for rotating components or free surfaces.

Performance Model Results

According to the performance model, the optimal cost efficiency for distributed-memory computers is obtained by choosing the minimum number of processors required to provide the necessary global memory, since this gives 100% memory efficiency and small communication/computation ratio. If necessary, this run time can be reduced by increasing the number of processors, although with reduced memory efficiency and increased communications overhead.

The model estimates for CPU, cost, and memory efficiencies are shown in Figures 1a and 1b for a modern but generic computer having the following parameters: effective CPU (*100Mflops*) and buffering (*30Mb/s*) rates, MPI bandwidth (*130Mb/s*) and latency (*15μs*), and memory of *512Mb* per processor.

These parameters are shown for both *memory-constrained sizeup,* in which the problem size is increased to maintain $\eta_{Mem} \approx 100\%$ as processors are added, and a *constant-problem-size scaleup* for 10 million grid points.

As expected, the CPU efficiency is higher for memory-constrained sizeup, but the difference in cost efficien-

Article Continues...

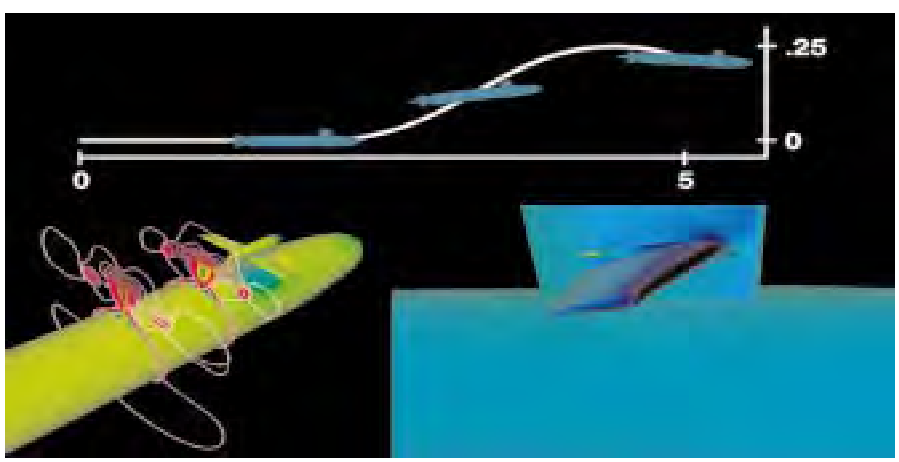


Figure 3. Rising Maneuver Induced by Sailplane Motion

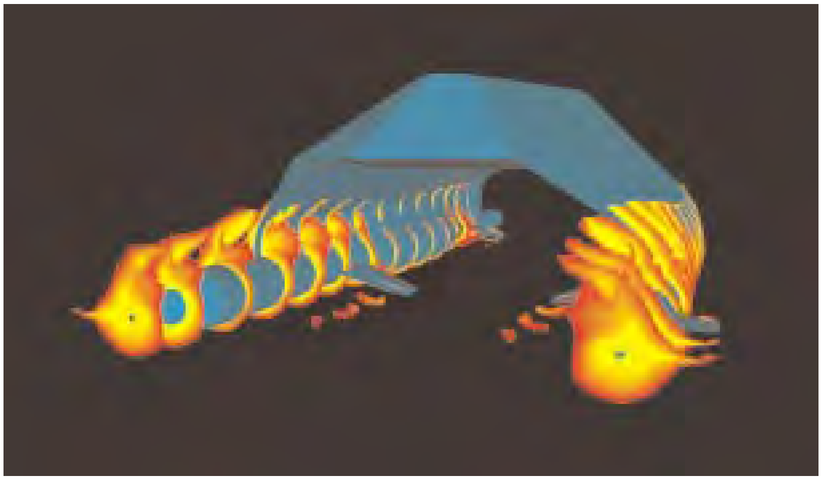


Figure 4. Solution with Free Surface for a Notional SWATH Hull Design Concept

cy is more dramatic due to the rapid drop in memory efficiency for constant problem size.

The most important computer parameters for scalability are effective CPU rate as timed for the executable operating with message passing suppressed, the MPI software bandwidth for large messages, and the time required to load message-passing buffer arrays. The MPI latency is negligible since there are only a small number of large messages.

Overall, the performance model indicates that the method is scalable in a practical sense for large-scale problems.

RECENT UNSTEADY SIMULATIONS

A number of large-scale simulations for complex geometries involving rotating components have been performed during the past two years. Figure 2 gives an example of a structured-grid solution for a propelled submarine configuration (SUBOFF) in straight-ahead motion, with a Reynolds number of 12 million. Figure 3 shows a rising maneuver induced by a prescribed motion of the sailplane control surfaces.

Shown are the trajectory in submarine lengths, a closeup view of surface pressure near the sailplane, and axial velocity contours revealing tip vortices behind the deflected sailplanes. This case has 4.5 million points, and each hull length traveled (2200 timesteps) requires 53 hours on 50 T3E processors (3.5 GigaFlop (Gflops), η_{CPU} = 87%). Figure 4 provides another structured-grid example for a surface ship solution with free surface.

Figures 5 shows an unstructured-grid solution for a notional submarine in straight-ahead motion at full-scale Reynolds number (i.e., 10^9). The sublayer resolution for this and other solutions here is such that $y^+ < 1.0$ at all surface points.

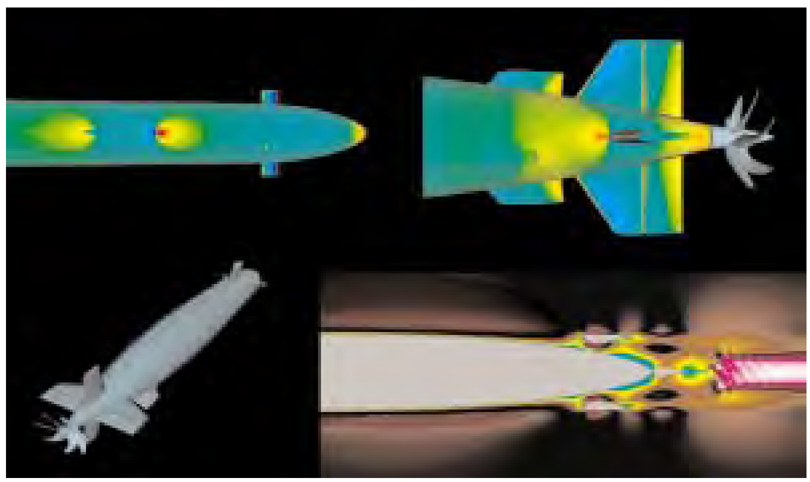


Figure 5. Notional Submarine at Full-Scale Reynolds Number(10^9)

Figure 6 shows a second unstructured-grid example for a Model 5415 destroyer hull that includes both rotating propellers and nonlinear free-surface conditions. This case, at Fr = 0.28, is especially difficult because of the wetted transom stern.

Figure 7 shows the computed maneuvering trajectory and axial velocity contours at two instants in time for a notional tilt-rotor aircraft solution, obtained from a combined unstructured grid simulation and 6DOF analysis. Figures 8a and 8b show a visualization of vortex concentrations behind a P5168 marine propeller.

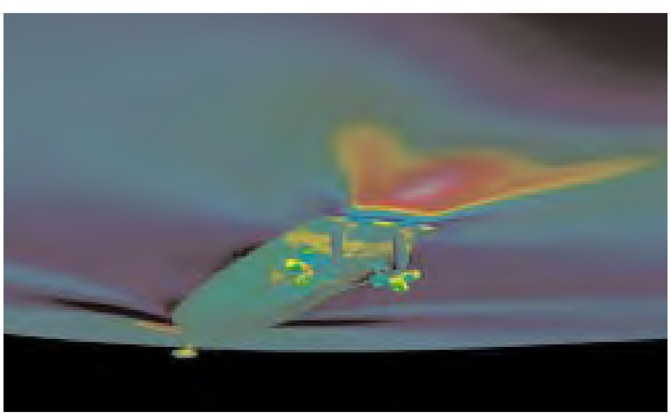
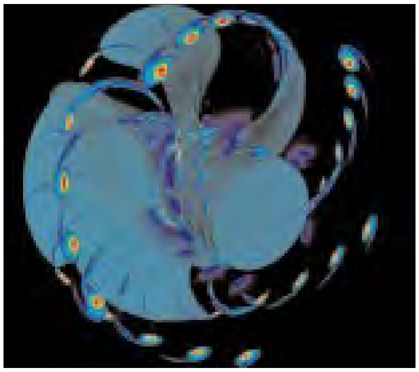


Figure 6. Model 5415 Destroyer Hull with Nonlinear Free-Surface Conditions

Finally, Figure 9 gives a validation comparison of both structured and unstructured solutions with experimental measurements for the P5168 propeller.

A high-resolution, time-accurate solution for a structured grid of one million points requires 2.1 Gigabit (Gb) of



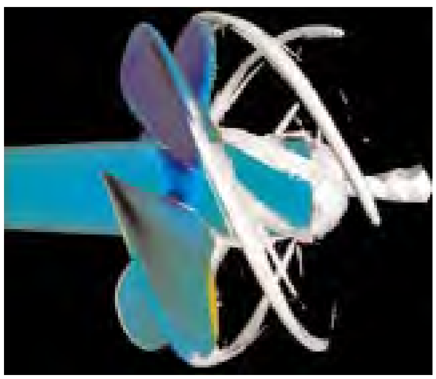


Figure 8a (above), 8b (right), Vortex Feature Detection in a Computed Flow for a P5168 Propeller

memory and about 0.25 processor hours per time step on a Sun ULTRA10000. The unstructured-grid code requires the same 2.1 Gb of memory, and although it requires more than double the run time per grid point, comparable viscous resolution has been achieved with 2-5 times fewer grid points with unstructured grids by exploiting local control of point distributions.

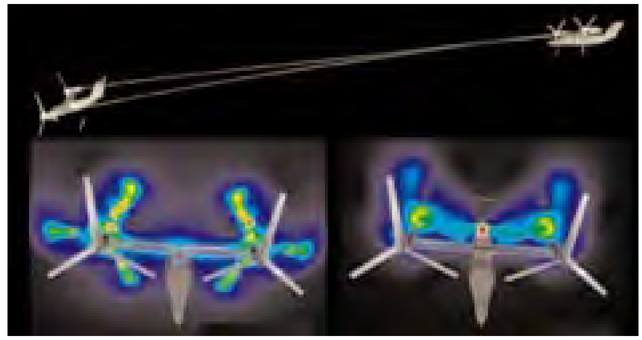


Figure 7. Computed Trajectory and Flow for a Tiltrotor Aircraft Maneuver Induced by a Sudden Wind Gust Following Ice Buildup on Wing Surfaces

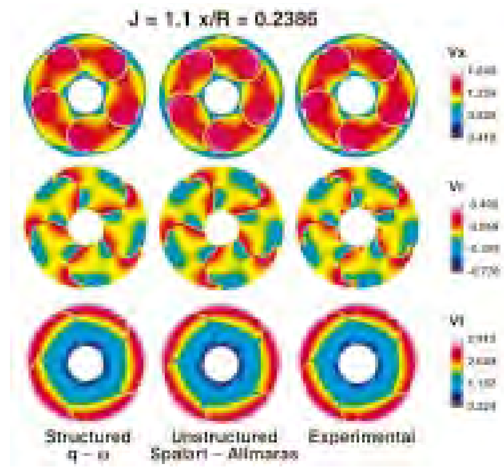


Figure 9. Validation of Structured and Unstructured Solutions with Experimental Measurements for a P5168 Propeller (Axial, Radial, and Circumferential Velocity)

References

1. Pankajakshan, R., Taylor, L. K., Sheng, C., Jiang, M., Briley, W. R., and Whitfield, D.L., "Parallel Efficiency in Implicit Multiblock, Multigrid Simulations, with Application to Submarine Maneuvering," AIAA Paper 2001-1093, 39th Aerospace Sciences Meeting, Reno, NV, January 2001.

2. Hyams, D. G., Sreenivas, K., Sheng, C., Briley, W. R., Marcum, D. L., and D. L. Whitfield. "An Investigation of Parallel Implicit Solution Algorithms for Incompressible Flows on Multielement Unstructured Topologies," AIAA Paper 2000-0271, 38th Aerospace Sciences Meeting, Reno, NV, January 2000.

Acknowledgements

This article includes examples of results from long-term team efforts of numerous researchers at the ERC SimCenter. In addition to the particular efforts of those who co-authored references 1-2, the efforts of M. Beddhu, C. O. E. Burg, B. Jayaraman, X. Wang, M. G.Remotigue, S. Nichols, J. M. Janus, J. Chen, K. Gaither, B. Mitchell, J.Newman, E. Blades, L. Massa, W. Brewer, A. Gaither, and H. Beeland have contributed strongly to the capabilities discussed herein. In addition, the two Challenge project teams include researchers from NSWC Carderock, University of Iowa, and SAIC. All visualizations shown here were prepared using the ERC SimCenter's DIVA visualization system.

Nanotubes

J. Bernholc, Department of Physics, North Carolina State University, Raleigh, NC

Carbon is unique among elements in its ability to assume a wide variety of different structures and forms. About fifteen years ago a new family of carbon cage structures, all based on a threefold coordinated sp^2 network, was discovered. This discovery inaugurated the science of fullerenes.

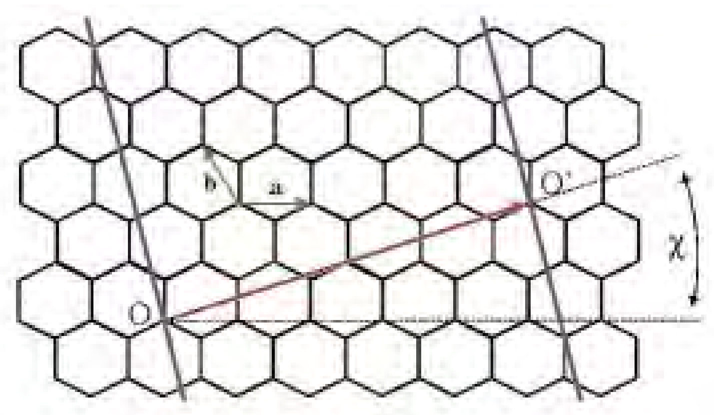


Figure 1a. **Nanotube structures are obtained by rolling a graphitic sheet into a cylinder. (*See Figure 1b*) The points O and O' in the graphite sheet fold into each other, and the nanotube structure is uniquely defined by the coordinates of the smallest folding vector (n,m) in the basis of lattice vectors a and b. The (n,0) zigzag and (n,n) armchair tubes are mirror-symmetric; all other tubes are chiral, i.e., the hexagon bands wind around the nanotube with a non-zero pitch.**

Of these, C_{60} is the most abundant and perhaps the best-known member. However, perhaps the most exciting among the recent additions to the fullerene family are carbon nanotubes, discovered soon after the C_{60} was made in quantity. Carbon nanotubes are hollow cylinders consisting of "rolled-up" graphitic sheets, as illustrated in Figures 1a and 1b. They are believed to have extraordinary structural, mechanical, and electrical properties that derive from the special properties of carbon bonds, their unique quasi-one-dimensional nature, and their cylindrical symmetry. For instance, the graphitic network upon which the nanotube structure is based is well known for its strength and elasticity, thereby providing for unmatched mechanical strength.

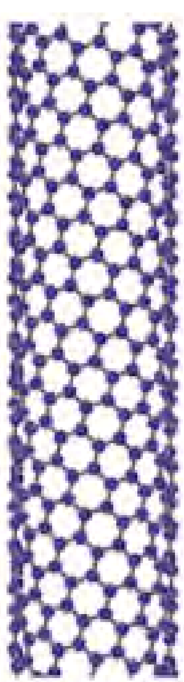


Figure 1b. **Nanotube structure obtained by rolling a graphitic sheet into a cylinder.**

Nanotubes can also be metallic or semiconducting, depending on their chirality (see Figure 1a). This opens up the very interesting prospects of junctions and devices made entirely of carbon.

The Strongest Materials Known

Nanotubes have very special mechanical properties. Their aspect ratios are enormous, with currently manufactured nanotubes having widths of 1-2 nanometers and lengths ranging from a fraction of a micron to a fraction of a millimeter. They can be thought of as fibers, which could be employed to strengthen composite materials—or used directly—when methods to grow long tubes in quantity are developed.

Computer simulations, which have been confirmed by careful nanoscale experiments, have shown that nanotubes are extremely flexible; they can bend reversibly to very high angles without exhibiting any damage even at an atomic scale. Supercomputer simulations have also predicted the immense strength of pure nanotubes, more than ten times the strength of steel at one-sixth the weight. Recent calculations by Qingzhong Zhao and Marco Buongiorno Nardelli at North Carolina State University suggest that the effective strength of nanotubes could even be significantly greater than that, because of the large "activation barriers" for atom rotation (see Figure 2), which must be overcome during breakage.

Single-walled nanotubes—consisting of a single cylinder—like to form bundles, or "ropes," while multiwalled nanotubes are made up of a number of concentric cylinders which do not necessarily have the same helicity, or pitch. Depending on the helicity, the nanotubes can be conducting, semiconducting, or insulating. Thus, they are excellent candidates for multifunctional materials, which provide both enormous strength as well as electrically conducting or insulating properties, as needed. However, controlled growth of nanotubes with desired length and pitch is still some time off.

When this is accomplished, even more interesting applications become possible, since calculations have shown that while some (armchair) nanotubes are conducting even when severely bent, other (chiral) metallic nanotubes lose their conductivity during bending, providing a nanoscale strain sensor.

NANOTUBE-BASED DEVICES

Nanotubes are excellent building blocks for nanoscale electronic devices. Due to their small dimensions and essentially perfect structure, a variety of novel devices become possible, including nano-electromechanical systems (NEMS), efficient electron emitters for flat panel displays and vacuum electronics, nanoscale chemical sensors, actuators, and even single-electron transistors. Some of these devices have already been realized experimentally, but many obstacles to their use still remain.

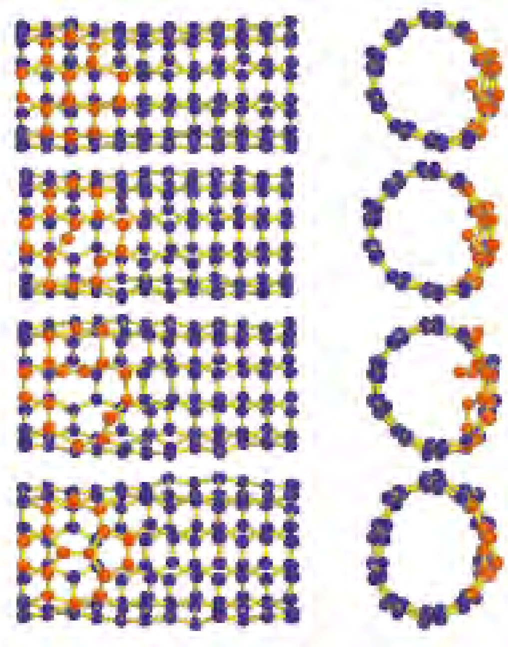


Figure 2. **Quantum molecular dynamics simulations show that nanotubes initiate breakage by a bond rotation, where a pair of atoms rotates about the center of their bond and converts four hexagons (highlighted in red) into a 5-7-7-5 defect. The barrier for this rotation is very high, which further increases the exceptional strength of nanotubes.**

In most cases, the overriding issue is controlled fabrication, but for quantum devices the underlying limitations must also be explored. One potential limitation is the huge megaohm resistance of nanotube-metal contacts, which is 100-1000 times more than expected. Such resistance could be due to poor fabrication in the experimentally very difficult nano-regime, but it could also have a fundamental physical origin.

Our group has thus embarked on a comprehensive study of nanotube-metal contacts by developing complex quantum-mechanical methods, which can compute the quantum transport properties of electrons of a nanotube-metal contact coupled to an external circuit. The required calculations are very demanding computationally, but the techniques developed as part of the Multiscale Simulations of Nanotubes and Quantum Structures project enable effective parallelization and therefore massively parallel execution.

The results of the first such calculation are shown in Figure 3, which depicts the electron distribution in a nanotube-aluminum contact and the transfer of electron charge between the two systems.

A sophisticated analysis of the results shows that the high resistance is caused by a fundamental reason, namely a "weak coupling" or a lack of common electron conductance channels between the perfect nanotube and the metal. However, there are several ways in which the coupling might be enhanced, including mechanical pressure on the contact region. The contact could thus be part of a nanoscale pressure sensor, and several other device configurations are possible.

Our current work focuses on more in-depth investigations of potential nanotube-based devices, and we are collaborating with Department of Defense-sponsored experimentalists at the University of North Carolina in Chapel Hill. Apart from NEMS-structures, we are also evaluating nanotubes for battery applications.

Early indications suggest that Li-nanotube batteries will have higher capacities and higher discharge and recharge rates than those based on graphite, but a lot of experimental and theoretical research still needs to be carried out.

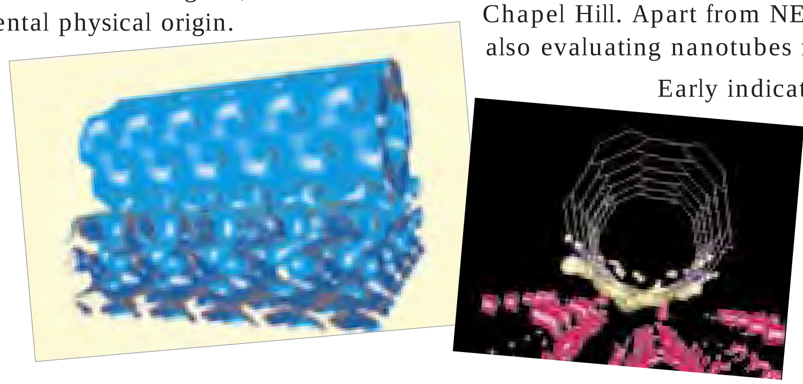


Figure 3. **The top panel shows the distribution of electrons in a carbon nanotube deposited on the surface of aluminum. The lower panel shows the charge transfer, namely the electrons that left the nanotube (purple) and entered the metal (blue). The calculations were performed by Marco Buongiorno Nardelli and Jean-Luc Fattebert.**

OpenMP Parallelization of a 3-D Finite Element Circulation Model

Dr. Timothy J. Campbell, NAVO MSRC Programming Environment & Training
Dr. Cheryl Ann Blain, Oceanography Division, Naval Research Laboratory

The Naval Oceanographic Office (NAVO) Major Shared Resource Center (MSRC) Programming Environment and Training (PET) program offers Department of Defense (DoD) researchers and engineers the opportunity to work in close collaboration with PET analysts to bring about serial and parallel optimizations to their applications.

One such collaboration occurred in a project that successfully ported an advanced three-dimensional (3-D) finite element (FE) circulation model to shared memory parallel machines.

The goal of this project was to produce, inasmuch as possible, a scalable code that required no change to the user interface and configuration files, while at the same time, to educate the researchers in parallel programming techniques.

OpenMP multithreading directives were chosen to port the model as they can provide a minimally intrusive and incremental method for producing a parallel code.

> "The goal of this project was to produce, inasmuch as possible, a scalable code that required no change to the user interface and configuration files, while at the same time, to educate the researchers in parallel programming techniques."

Physical & Mathematical Model

Parallelization efforts were focused on the Dartmouth College circulation model, QUODDY, which represents the most physically advanced finite element model to date.

This model is a time-marching simulator based on the 3-D hydrodynamic equations subject to the conventional Boussinesq and hydrostatic assumptions. A wave-continuity form of the mass conservation equation, designed to eliminate numerical noise at or below two times the grid spacing, is solved in conjunction with momentum conservation and transport equations for temperature and salinity.

Vertical mixing is represented with a level 2.5 turbulence closure. This turbulence closure scheme accounts for processes occurring over the vertical extent of the water column, such as diffusion, shear production, buoyancy, production, and dissipation. Variable horizontal resolution is provided on unstructured triangular meshes. A general terrain-following vertical coordinate allows smooth resolution of surface and bottom boundary layers.

The QUODDY model is dynamically equivalent to the often used Princeton Ocean Model. The advantage of the current model lies in its finite element formulation that allows for greater flexibility in representing geometric complexity and strong horizontal gradients in either bathymetry and/or velocity.

Parallel Implementation

OpenMP is a parallel programming model for shared memory and distributed shared memory multiprocessors that works with either standard Fortran or C/C++.

OpenMP consists of compiler directives, which take the form of source code comments, that describe the parallelism in the source code. A supporting library of subroutines is also available to applications. The OpenMP specification and related material can be found at the OpenMP web site: http://www.openmp.org.

Online training in OpenMP is part of the NAVO MSRC PET distance learning (http://www.navo.hpc.mil/pet/Video) and links to other online training material can be found at the NAVO PET Parallel Computing Portal (http://www.navo.hpc.mil/Tools/pcomp.html).

In Fortran, OpenMP compiler directives are structured as comments, written as C$OMP or !$OMP. An OpenMP program begins as a single process, called the master thread. When a parallel region, which is preceded by either a parallel or parallel-do construct, is encountered, threads are forked to execute the statements enclosed within the parallel construct.

At the end of the parallel region, the threads synchronize, and only the master thread remains to continue execution of the program. The parallel-do construct is commonly discussed and provides a convenient and incremental way to parallelize computationally intensive loops within a program.

The downside to this approach is that the creation of threads at the beginning and their subsequent destruction at the end of the parallel-do construct can require a large number of cycles. The developer must be sure that the loop being parallelized has enough computational

work to make the overhead, due to the OpenMP constructs, worthwhile.

The approach used in this project is in the spirit of the Single Program Multiple Data (SPMD) model which is common in Message Passing Interface (MPI) programming. The parallel/end parallel directives were used to enclose the entire time-stepping portion of the code, including subprogram calls within the parallel execution region. Work decomposition within the parallel region is based on the horizontal mesh.

During execution in the parallel region, the threads remain in existence, and proper data flow is ensured through minimal use of the barrier synchronization construct. Also, code that must be executed in serial is handled by the master thread.

Since the barrier construct can be 30 to 50 percent less expensive than a parallel do, this approach significantly reduces the amount of overhead associated with OpenMP.

The QUODDY software model consists of four sets of programs and includes files for the dimensioning of variables. Parallelization work focuses on three of the program sets that consist of main, core, and fixed routines. When a user applies the QUODDY application to a particular regional model, these three sets of programs remain unmodified. The fourth program set consists of user-built subroutines that are built with a standardized interface.

These routines are used to specify things such as physical forcing, vertical meshing, boundary conditions, and the manner in which results are to be analyzed and written.

By restricting the OpenMP code changes to the main, core, and fixed routines, the user is able to seamlessly apply the parallel QUODDY to different regional models. The user need only compile with the subroutines defined for the regional model of choice.

Verification & Proper Performance

Correctness of the parallel code execution has been verified through direct comparison with the original serial code execution for the Yellow Sea Regional Model (6847 horizontal & 21 vertical nodes).[1] This verification was done using the full "seasonal" mode in which wind is applied and temperature and salinity are transported prognostically.

Since the user-defined output data was of limited precision, verification was done by directly comparing (at full precision) all time-integrated variables. Possible race conditions were "fleshed out" by running with the number of threads greater than the number of processors. An exact match between the serial and parallel execution has been achieved.

Performance measurements were done using the Arabian Gulf regional model (17440 horizontal nodes and either 21 or 51 vertical nodes).[2] The speed-up on p processors is defined as the single processor execution time divided by the time for execution on p processors. Figure 1 shows the speed-up achieved for the OpenMP version of QUODDY on the NAVO MSRC Sun E10000 (64 processors with 64 Gigabyte (GB) shared memory).

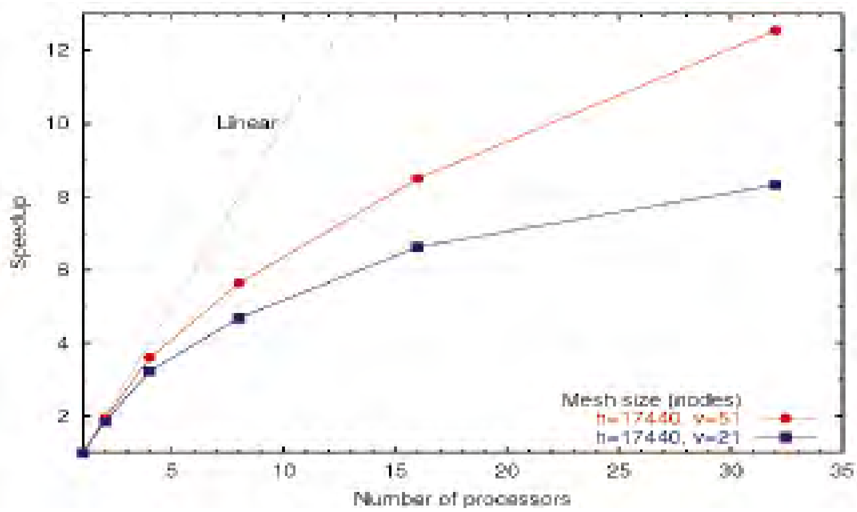


Figure 1. **Speed-up of OpenMP QUODDY4 for the Arabian Gulf (17440 horizontal nodes) on the NAVO Sun E10000. Results for two vertical mesh resolutions are shown: 21 vertical nodes (blue-filled squares) and 51 vertical nodes (red-filled circles).**

Two vertical grid resolutions (21 and 51) were measured. The increase in vertical grid resolution provides more work per horizontal node, thus increasing the scalability of the code. The overall scalability of the OpenMP QUODDY is limited by the remaining serial portions of work (about 5 percent, handled by the master thread) and the synchronization overhead.

Impact & Application

The state-of-the-art QUODDY 3D FE model is a principal tool in the NRL Arabian Gulf project, of which, the primary objective is development of a circulation model for the Arabian Gulf and connecting waters that realistically predicts the complex 3-D circulation and mixing patterns in the region over seasonal, tidal, sub-tidal, and storm event time scales. Mesh resolution is variable, approximately 3 kilometer (km) for depths less then 40 meters (m) and 6 km elsewhere out to 200-m depth in the Gulf of Oman.

Article Continues Page 14...

U.S. West Coast

Researchers at the Naval Postgraduate School, Monterey, California, in conjunction with the NAVO MSRC Visualization Center staff, have undertaken a long-term project to improve the performance of the Multiblock Grid Princeton Ocean Model (MGPOM). The use of multiblock grids in the development of ocean models facilitates domain composition and varying grid resolutions to provide the ability to concentrate grid resolution in the dynamic near-shore regions and save resolution in the less dynamic deep-ocean areas.

Traditional one-block rectangular grids (286 x 286, 4 arc minute resolution), while invaluable, consume large quantities of wall time, slowing research and raising costs. For example, a traditional single block, serial (vector) code takes approximately 1,116 minutes (18.6 hours) of wall time to complete a 10-day simulation. In comparison, a 29-block grid with the same resolution, using MPI-Pthreads MGPOM code, takes only 27 minutes of wall time.

The model produced with this new and improved code provides three-dimensional (3-D) temperature, salinity, and circulation (currents) data as shown in Figures 1 through 5. These images represent screen captures of an analysis environment built for these researchers by the NAVO MSRC Visualization Center staff. This application, and others developed by the Visualization Center staff, provides researchers with a portable analysis environment for ocean model output that supports a variety of functions for both the military and civilian communities.

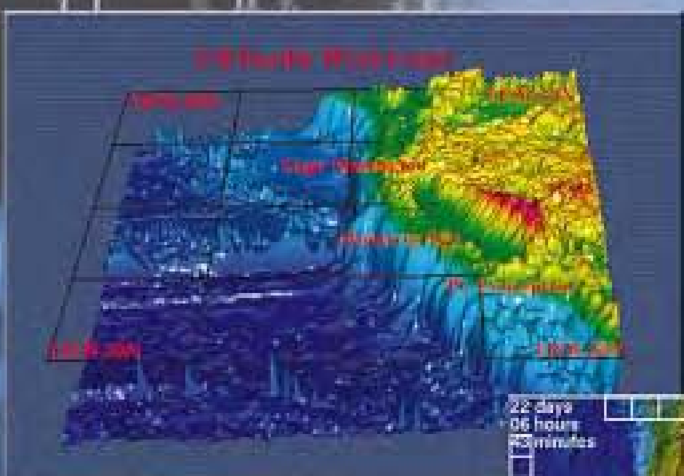


Figure 1. **Plain view of U.S. West Coast with multiblock grid.**

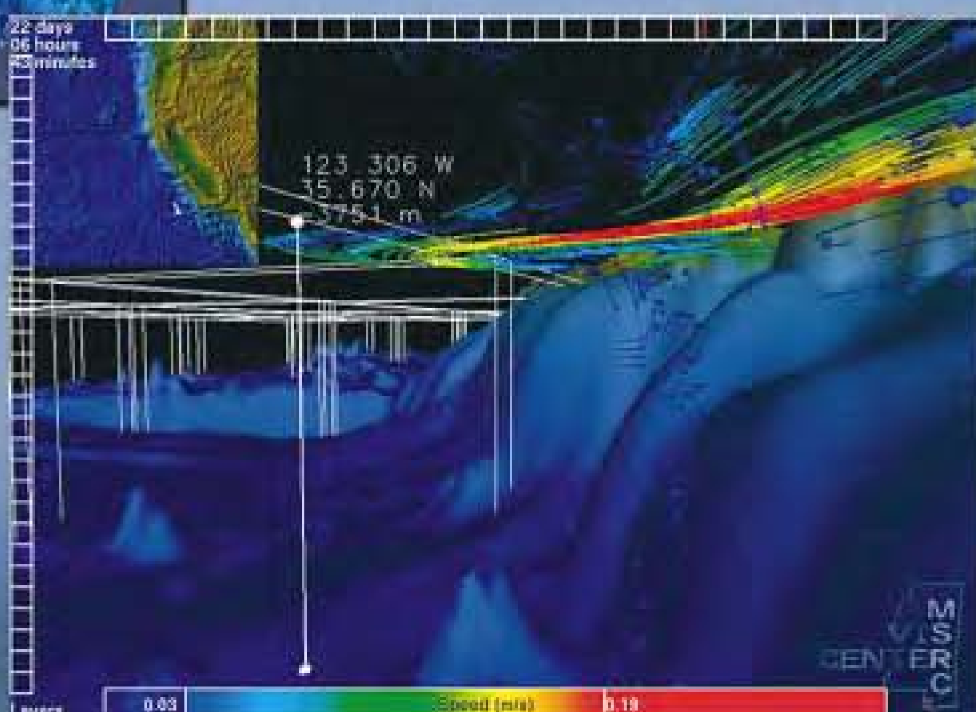


Figure 2. **Perspective view showing the streamlines of surface currents off the U.S. West Coast.**

c Ocean Currents

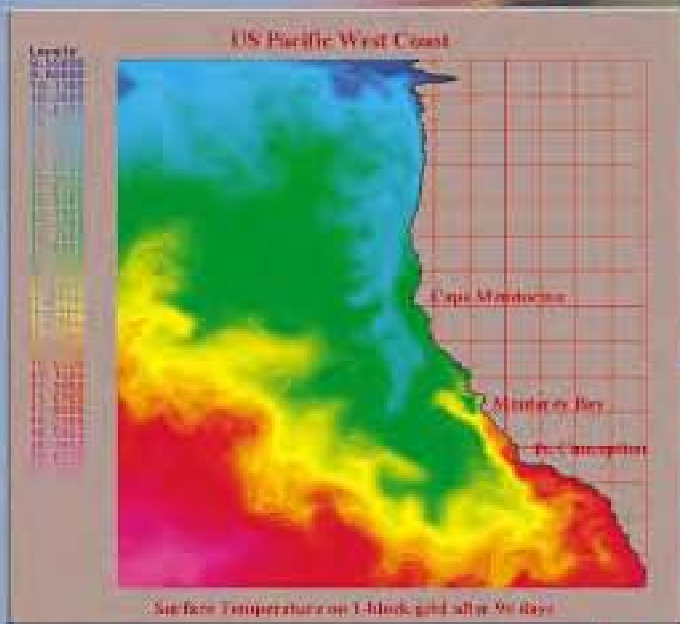


Figure 5. **Surface temperatures after a 90-day simulation within a one-block grid.**

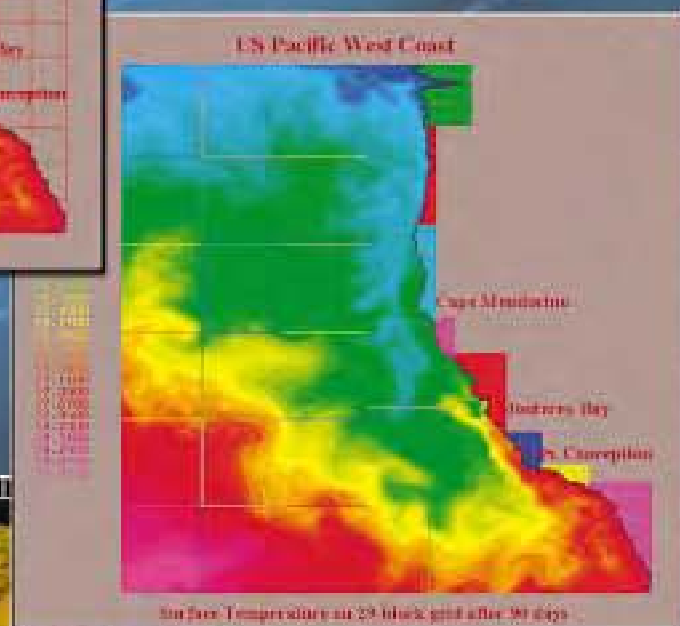


Figure 4. **Surface temperatures after a 90-day simulation within a 29-block grid.**

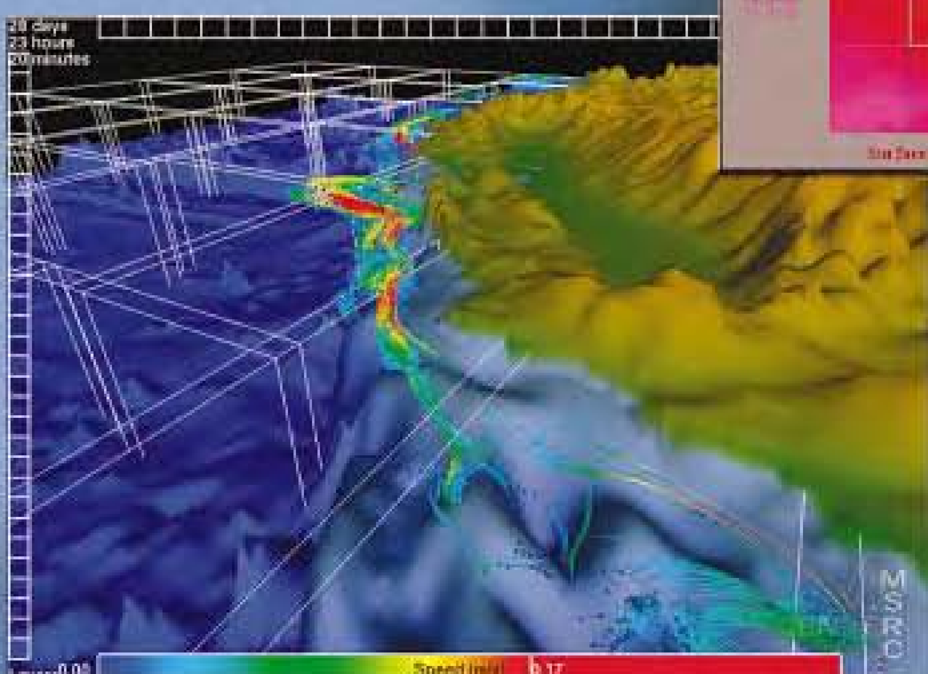


Figure 3. **Top-down view showing the streamlines of surface currents off the U.S. West Coast.**

Continued from Page 11...

The model is designed with modular dynamics in which certain mechanisms, such as heat flux, wind forcing, stratification, tides, or river inflow can be independently included or excluded from model equations. This modularity is used to examine the contributions of each component to the overall circulation dynamics.

Simulations are forced by seasonal hydrography, seasonal winds, and tides. For each month, the initial temperature and salinity fields prognostically evolve subject to tidal rectification and a constant wind stress.

The summer mean circulation is primarily driven by the baroclinic pressure gradient. Fresh water entering the Arabian Gulf from the Gulf of Oman at the surface, coupled with strong evaporation in the north, creates a cyclonic circulation gyre that runs the length of the basin. The northwesterly wind strengthens southward flow along the western edge of the gyre. A westward component of the wind in the southern Arabian Gulf pushes water across the very shallow shelf of the United Arab Emirates (UAE) coast and out through the Strait of Hormuz (Figure 2a).

During winter, the strong northwest winds (3 times the magnitudes in summer) set-up southeastward flowing coastal currents in the northern Gulf along each shoreline. The winds also impede penetration of the freshwater into the Gulf and greatly reduce the strength of the counter-clockwise (CCW) circulation along the axis of the basin (Figure 2b).

In fact, the winds push the circulation gyre to the south and toward the center of the Gulf. Since there is no westerly component to the wind in winter, the circulation on the shallow southern shelf is quite complex and varied from that seen in summer.

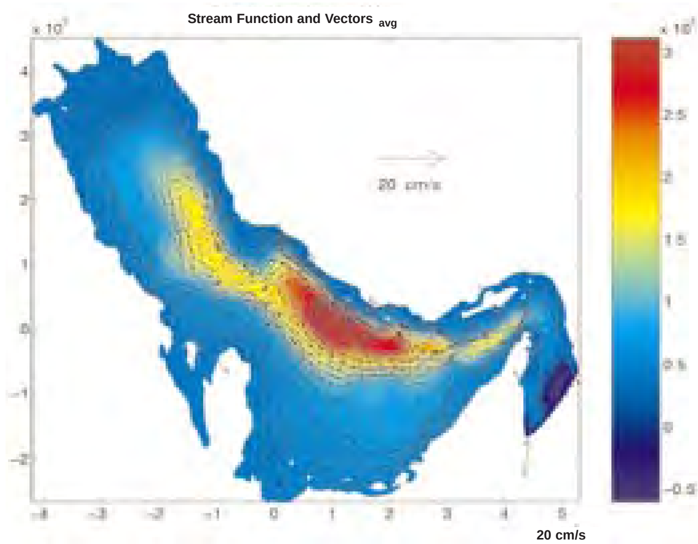


Figure 2b. **Simulated winter seasonal circulation in the Arabian Gulf. Stream function (color) and depth-averaged currents (vectors).**

The speed-up achieved by the OpenMP version of QUODDY is immediately useful to the Arabian Gulf and other planned modeling work. Prior to porting QUODDY model, it would not execute properly on the MSRC resources, thus restricting the researchers to perform simulations only on their workstations.

Performing 10-model-day seasonal simulation experiments required up to several days of execution with limited vertical grid resolution. Now, on 8 processors of the NAVO MSRC Sun E10000, researchers can perform the same 10-model-day seasonal simulation with increased vertical grid resolution in just over 6 hours (with 71 percent parallel efficiency).

The reduced turnaround time will greatly accelerate the model development process. Additionally, because this was a collaborative effort, the researchers are now familiar with the OpenMP code changes and are able to modify and improve the parallel code.

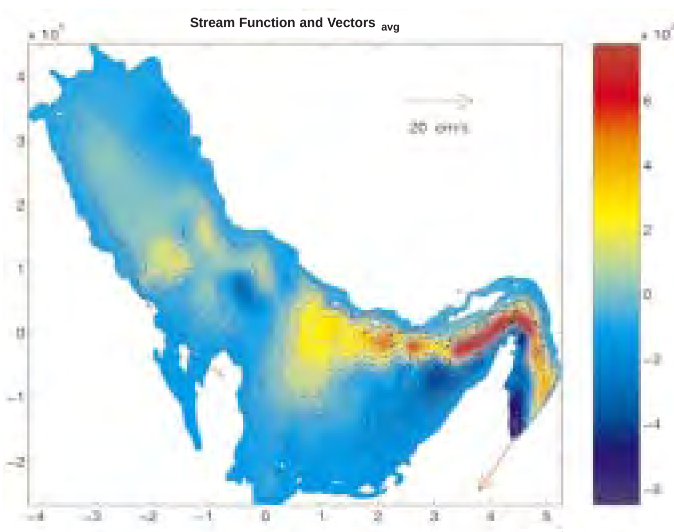


Figure 2a. **Simulated summer seasonal circulation in the Arabian Gulf. Stream function (color) and depth-averaged currents (vectors).**

References

1. C. E. Naimie, C. A. Blain, and D. R. Lynch, 2000. Seasonal mean circulation in the Yellow Sea—A model-generated climatology, Continental Shelf Research, in press.

2. Blain, C. A., 2000. Modeling three-dimensional, thermohaline-driven circulation in the Arabian Gulf, in Estuarine and Coastal Modeling, Proceedings of the Sixth International Conference, M. L. Spaulding and H. L. Butler, eds., American Society of Civil Engineers, pp. 74-93.

Visualization Center Video Production Studio Ready to Take on 21st Century in Style

Kerry Townson, Multimedia Specialist

Left: **The main console houses two nonlinear editing systems and their host computers, audio processing and monitoring equipment, user-assignable video monitors, and a touch panel to perform signal routing tasks.**

With an eye to the future and an emphasis on flexibility, the Visualization Center Video Production Studio (VPS) was recently upgraded to better fulfill the audio-visual requirements of the NAVO MSRC.

Created eight years ago to produce videotaped programs about the MSRC and to support its research activities, the VPS now boasts two separate nonlinear editing systems, extensive graphics capabilities, and the ability to quickly produce multimedia products for a variety of uses. Besides the traditional videotape output, the studio can produce materials for multimedia presentations on CD-ROM, webcasting, and Digital Video Discs (DVD).

Going Digital

Like many technical fields, video production has moved to the desktop. The original design of the VPS was similar to a broadcast facility, with videotape machines feeding a rack full of discrete components. Each piece of equipment handled one particular task such as video titling, digital special effects, audio production, or automated editing. Routing tape machines to different pieces of equipment required attaching cables by hand or using an awkward patchbay device. The system's complexities often left the video editor wishing for extra hands and reduced the efficiency and expediency of project completion.

Above: **The main rack contains six video sources, a dubbing station, routing hardware, and the base station for one of the nonlinear editors. The studio can accommodate eight different videotape formats and a variety of digital files.**

Advances in technology have combined most of the tasks into a desktop computer environment that enables an operator to manage several tape decks, special effects, and audio with a few clicks of a mouse button. A sophisticated routing system controlled by a touch panel allows the signals of any tape deck, monitor, or editing system to be fed to any other device with just three taps on the screen.

Digital video has brought the ability to work entirely within the computer, with pristine quality and amazing special effects that were impossible only a few years ago without hundreds of thousands of dollars worth of broadcast equipment. Video can now be fed to the system either in digital format directly from the NAVO MSRC super computers or encoded from analog videotapes. Stored in digital format on the computer's hard drives, the video can be fed back to tape machines for duplication, saved to digital files, and distributed via disk, CD-ROM, DVD, or written to a network computer for future use.

The Hardware

At the heart of the system is a computer-based routing system that handles all of the video, audio, synchronization,

Article Continues...

Right: **Audio processing equipment includes Digital Audio Tape (DAT), CD-ROM, multiple audio effects units, and a computer-based Digital Audio Workstation.**

and control signals from every component. A touch panel makes it simple to connect any machine's outputs to another's inputs, or to a monitor or editing system.

The VPS now has two nonlinear editing (NLE) systems. The Trinity NLE is used primarily for high-quality video. It features all-digital editing of extremely high-resolution images with no loss of quality during the editing process, as well as a collection of eye-popping special effects. Video clips are dragged onto a timeline and combined with graphics, special effects, and sound to create polished, professional programs on par with those seen on commercial television.

A second NLE is dedicated to multimedia applications. Based on a Matrox RT2000 card installed in a standard PC, this system is used for video projects that will be inserted into CD-ROMs or PowerPoint presentations.

The VPS can also create video footage for its clients. A professional 3-chip digital camcorder is available for shooting on location. A sound booth, 16 channel audio mixer, dual audio effects units, CD-ROM player, and a digital audio tape (DAT) recorder round out the audio capabilities of the new system.

APPLICATIONS

The VPS provides a wide variety of services to the NAVO MSRC and its clients. The VPS has provided short

Above: **A dedicated sound booth provides the ability to record high-quality narration for multimedia projects.**

video clips and voiceovers for use in two CD-ROMs distributed at super computing conventions and in PowerPoint presentations.

Visualization Center animators use the facility to assemble high-definition renderings of computer data generated by the super computers of the MSRC. A network interface allows single frames of animations to be fed to the VPS, assembled into a video program, and recorded onto disk or videotape. Real-time capture of super computer displays permits the transfer of interactive simulations such as Theater High Altitude Area Defense (THAAD) and Miami Isopycnic Computer Ocean Model (MICOM) to videotape.

Other projects include a video introduction to the NAVO MSRC for the numerous grade-school classes who visit the facility each year and a more detailed documentary of the center's activities aimed at adult visitors. VPS projects are viewed daily on large-format displays in the Stennis Space Center Visitors Center and outside the NAVO MSRC Visualization Center.

Left: **Electronic News Gathering (ENG) equipment adds on location videotaping to the studio's capabilities.**

With its state-of-the-art equipment and enhanced capabilities, the VPS staff looks forward to providing a new level of service to the NAVO MSRC and its clients needing audio-visual support.

SUPERCOMPUTING 2001

DENVER, COLORADO

SC2001

Nov. 10-16 2001

April 9 - Submissions on Website opens **April 27** - Submission deadlines for Gordon Bell Awards **May 28** - Education Program Applications Due **June** - Conditional Acceptance for Technical Papers **July 30** - Final Version of Technical Papers due to Confirm Acceptance Student Volunteer Applications Due Submission deadline: Exhibitor Forum HPC Games - Posters **August 31** - Final Tutorials Handouts Due **September 7** - Birds of a Feather (BOFs) Submission Deadlines **November 10** - SC2001 begins

www.sc2001.org

NAVO MSRC PET Update

Eleanor Schroeder, NAVO MSRC Programming Environment and Training Program (PET) Government Lead

The end of PET as we all know it is approaching, and we are spinning up an exciting all-new PET.

As the PET program evolved over the past five years, it took on a more global, more user-oriented, focus. While the original intent was to improve the productivity of users at the Major Shared Resource Centers (MSRCs), the new PET will expand to incorporate users located at the Distributed Centers and Department of Defense (DoD) remote locations.

There will still be four PET components, co-located at each MSRC. Each component is now responsible for specific functional areas as designated in the box below.

We have grouped these designated functional areas to encourage synergy among related Computational Technology Areas (CTAs), collaboration and interaction between CTA and cross-community functions, and to balance workloads. There are a total of fifteen functional areas, ten of which support the ten established CTAs. The other five are:

Collaborative and Distance Learning Technologies

This functional area encompasses not only virtual meetings (meetings without travel), but also technology for on-line training, consultation, information, and tutorials. The activities within this function are expected to interact with our training content providers for the development, testing, and deployment of distance learning technology and course material. We expect strong interaction with the Defense Research and Engineering Network (DREN) initiative to ensure coordination and incorporation of collaborative and distance learning technology into the High Performance Computing Modernization Program (HPCMP) networking and security infrastructure.

Four PET Components

Component 1 (NAVO): CWO/EQM; Computational Environment

Component 2 (ASC): FMS/IMT/SIP; Enabling Technologies

Component 3 (ERDC): CFD/CSM; PET Online Knowledge Center; Education, Outreach, and Training Coordination

Component 4 (ARL): CCM/CEA/CEN; Collaborative and Distance Learning Technologies

Computational Environment

Critical to easy and effective use of DoD High Performance Computing (HPC) systems resources, from the high performance computer down to the desktop, is improving the usability of computational environments at the DoD Shared Resource Centers (SRCs).

These computational environments encompass all aspects of the user's interface to HPC resources, including programming environments (e.g., debuggers, libraries, solvers, higher order languages, and performance analysis, and prediction and optimization tools), computing platforms (e.g., common queuing, clusters, distributed data, and metacomputing), reusable parallel algorithms, and user access tools (e.g., portals and web-based access to HPC resources).

Enabling Technologies

This functional area involves advancing the state of tools, algorithms, and standards for generalized run-time and pre- and post-processing analysis on enormous datasets. At a minimum, this function will entail visualization, data mining and knowledge discovery, image analysis, grid generation, problem-solving environments, and computational techniques and methods for the intelligent extraction of useful information from data.

PET Online Knowledge Center (OKC)

The OKC will provide repositories for PET programmatic information and technical knowledge in both the computational science and computational technology areas. It will also provide ready access to software tools and products, as well as current information on PET projects in all functional areas. Additionally, the PET OKC will allow HPCMP users and personnel to enter a single Web portal with one navigational hierarchy, information strategy, and search mechanism, to better allow them to distinguish vast amounts of information and expertise from distributed sites.

Education, Outreach, and Training Coordination

Under this functional area we will address the efficient and productive delivery of instructional content to the DoD HPC user as well as opportunities for Minority Serving Institutions (MSIs), undergraduate, graduate, and postdoctoral students, and visiting scientist/engineer appointments. Also, this is where we will tend to the training of future DoD HPC users.

The education of both novice and experienced HPC users in new and innovative technologies is an essential element of this functional area. While instructional content and delivery technologies are addressed in other PET functional areas, activities in this functional area will include coordination of on-site training at the SRCs and remote sites, selection of optimal training delivery methods and media, and coordination of outreach forums.

At NAVO MSRC, we look forward to this new version of PET. We believe it will bring us many new, exciting challenges and provide us with closer ties to the DOD HPC user community in the coming years.

On a personal note: As we close the door on the last year of PET as we know it, I would like to take a moment to thank the many academics that have worked with, and continue to work with, our NAVO MSRC team. Without your hard work and efforts, we certainly would not have had as successful a program as we did. I hope that our paths will cross again during this new evolution of PET.

PET Training - Practical and Hands On

Dr. Timothy J. Campbell, NAVO MSRC PET

Open MP Language and KAP Pro Tools for OpenMP "Bring Your Own Code" Workshop

In October and November 2000, the Naval Oceanographic Office (NAVO) Major Shared Resource Center (MSRC) Program Environment and Training Program (PET) held a four-day "Bring Your Own Code" workshop on the OpenMP language and the KAP Pro Toolset. Seventeen people attended the workshop in the NAVO MSRC PET classroom facilities, where all code development and application runs were performed on the NAVO MSRC Sun E10000.

The workshop was designed for experienced Fortran 77/90/95 programmers who have used serial platforms ranging from workstations to mainframes. However, no prior knowledge of programming parallel computers was assumed. While Shared Memory Parallel (SMP) platforms were the workshop target, special attention was devoted to issues related to porting legacy code to SMP OpenMP implementations.

The first two days consisted of an extensive OpenMP language course complete with hands-on exercises. On the third day, students received training on the KAP Pro Toolset for OpenMP provided by an instructor from Intel. Students were invited to "bring your own code" on the fourth day in order to receive direct assistance from the instructors in parallelizing their code using OpenMP and the KAP Pro Toolset.

The student-provided codes represented Climate/Weather/Ocean modeling (CWO) applications ranging from sediment transport, to acoustics, to wave modeling. In a short amount of time several students were able to parallelize computationally intensive loops within their applications and achieve a speed-up of about a factor of 2 on multiple processors. Typically, students were able to capture about 50 to 60% of the computation in parallel with about one to two hours of work.

NAVO MSRC PET would be pleased to repeat this workshop in the future. If you would like to attend an OpenMP or parallelization workshop in the future, please contact the NAVO MSRC PET Training Coordinator, Brian Tabor, at taborb@navo.hpc.mil.

IBM ACTC Applications on the IBM SP2 "Bring Your Own Code" Workshop

In support of user transition to the NAVO MSRC 1,336-processor IBM SP2, NAVO MSRC PET hosted a "Bring Your Own Code" workshop in December 2000. Experienced instructors from the IBM Advanced Computing Technology Center guided 12 attendees through the workshop.

The three-day workshop provided an opportunity for NAVO MSRC users to learn about developing and running applications on the IBM SP. A detailed introduction was given on hybrid distributed/shared cache-based parallel processors with a focus on the IBM Power 3 Winterhawk and Nighthawk nodes. Programming techniques for optimal uni-processor performance were presented, including cache utilization, stride elimination, and prefetching. Useful performance analysis and debugging tools were discussed and demonstrated. The morning sessions of days two and three covered programming techniques such as Pthreads, OpenMP, and Message Passing Interface (MPI) for shared and distributed memory parallelization.

Students were invited to bring their own codes for the day two and three afternoon sessions, which were devoted to the conversion and optimization of student codes. The instructors provided direct assistance to students from several Challenge and non-Challenge projects who brought codes representing the Computational Fluid Dynamics (CFD), Climate/Weather/Ocean Modeling (CWO), Environmental Quality Modeling (EQM), and Computational Chemistry and Materials Science (CCM) Computation Technology Areas (CTAs). For all of the attendees, the "bring your own code" session was time well spent. One student was able to resolve several debugging issues in a hybrid MPI/OpenMP wave modeling application. Another student made significant progress in porting a CFD application to MPI.

If you are interested in more information about the NAVO IBM SP, visit http://www.navo.hpc.mil/usersupport/IBM.

2001 Winter Applied MetaComputing/University of Virginia Legion Workshop

The joint Legion Group/Applied Metacomputing Winter 2001 Workshop was held in January at the University of Virginia. The workshop targeted all levels of users and administrators, and included information on customizing and troubleshooting Legion systems. Participants were introduced to the Legion system, philosophy, and architecture and were given an in-depth user's point of view. Hands-on sessions gave users the opportunity to adapt and run either their own or a test application in Legion, while system administrators had the opportunity to advanced administration topics such as building a system, adding resources to an existing system, and managing security.

For more information on the Legion workshops, visit http://www.legion.virginia.edu/workshops.html.

Navigator Tools and Tips

So You Were Given Hours to Run Your Model on Something Called the SV1

Ray Sheppard, NAVO MSRC User Support

That machine has four host names: Zeus, Poseidon, Trident, and Athena. How do you choose which "machine" to run on? You've got a 500-MB input data file and don't have the time to wait on pulling it off tape from the mass storage after your job starts running. You will load it into my/tmp directory first. You put it in /tmp/my_log_name on Athena, submitted your job, but got an error file that says "no such file or directory. The job tried to run on Zeus! How do you stop that?

Once you submit your job to the queue, normal users do not have control over where they are going to execute the job. At the moment, this is only a minor inconvenience because with only four nodes (machines), your 500-MB data file could be copied into four different /tmp directories, and you would be good to go. However, this machine has the ability to grow to 32 nodes and that would make pre-staging data a bit of a chore.

So, what is the good news? Well, you can still accomplish the pre-stage with only two data transfers, and it will not matter how many nodes the SV1 becomes. The trick is to pick a /tmp node that you would like to start from and copy your files there. Then you can submit your job which should begin by running a simple script to first test its environment, and then copy the /tmp environment from your node of choice to the node that has been selected for your job to run. This is only a minor delay since inter-node transfers are quickly performed. Your 500-MB data file should move in less than a minute (see statistics on a 49-MB file below).

Here are a few notes concerning this script:

Note 1: You should have a file in your home directory called ".rhosts". This file should be amended to include all of the nodes with a "-hip0" extension. An example would be:

```
athena
athena-hip0
athena.navo.hpc.mil
athena-hip0.navo.hpc.mil
trident
trident-hip0
trident.navo.hpc.mil
trident-hip0.navo.hpc.mil
zeus
zeus-hip0
zeus.navo.hpc.mil
zeus-hip0.navo.hpc.mil
poseidon
poseidon-hip0
poseidon.navo.hpc.mil
poseidon-hip0.navo.hpc.mil
```

Obviously, this file should grow as new nodes are added...

Note 2: This script is written in C-shell, but it may be called by other types of shells. If you do not like C-shell, I am certain that comparable Bourne, Korn, or shell of choice could be written.

Note 3: This script may be run embedded in your QSUB job or as an executable from your home directory.

Note 4: Finally, this script is going to look for a small source code file and a 49-MB data file in /tmp/ray on the node Athena. The script will time the transfers (the example ran on Zeus), compile and run the code, and cat its contents. (This code shows the accumulated error caused by summing the same number set forward and then in reverse.)

The Script (called news.host.csh)

```
#!/bin/csh
set echo
setenv HOST `hostname`
echo $HOST
if ($HOST != "athena")then
   if (! -d
/tmp/ray/round_error)then
      mkdir -p /tmp/ray/
round_error
   endif
   cd /tmp/ray/round_error
# Use the HiPPI connection
for the fastest internal
transfer speed.
   timex rcp athena-hip0:
/tmp/ray/round_error/input.f
.
   timex rcp athena-hip0:
/tmp/ray/round_error/data.d
at .
else
   echo "I do not need to
move files, so do nothing
here & go to work"
   cd /tmp/ray/round_error
endif
#
pwd
f90 -o test.job input.f
chmod 755 test.job
ls -l
ja
./test.job
ja -st
#
echo " End of Job "
#
```

Article Continues Page 22...

A Look Inside NAVO

We welcome our visitors...

Left:
Simone Youngblood, Defense Modeling and Simulation Office's Verification, Validation, and Authentication Technical Director visit

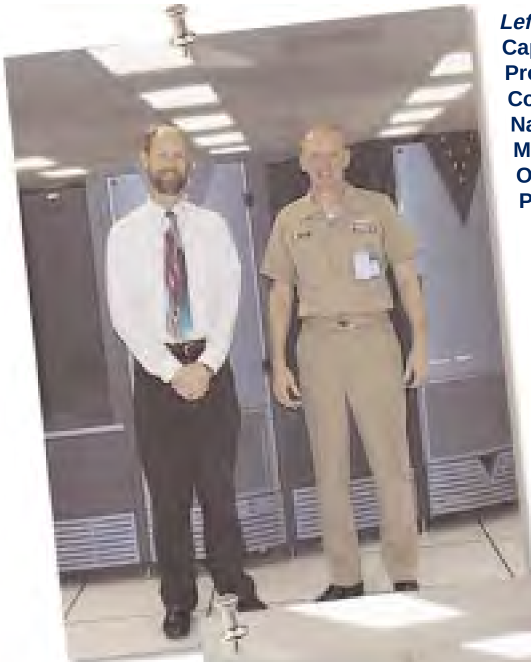
Right:
Captain Frank Garcia, Office of the Secretary of Defense visit

Left:
Congressional Delegation visit

Right:
National Imagery and Mapping Agency Delegation visit

Above:
Commemorative Keepsake Scientific Computing 2000 (SC2000). David Stinson, Charles Ray, and Dana Allen, Engineer Research and Development Center, Pete Grusinskas and Eleanor Schroeder, NAVO MSRC

Right:
Brigadier General George Cannellos, Adjutant General for Air, State of Alaska, visit



Left:
Captain Grandau, Prospective Commanding Officer, Naval Pacific Meteorology and Oceanography Center, Pearl Harbor, visit

Right
Colonel Robert Allen, Air Force Weather Agency, visit

Right:
Terry's Last Tour - L-R
Steve Adamec, NAVO MSRC Director; Dr. Donald Durham, CNMOC Technical Director; Terry Blanchard, NAVO MSRC Deputy Director; Landry Bernard, NAVOCEANO Technical Director

Above:
Terry in retirement

Above: **Captain Gunderson, NAVSEA visit**

Left:
Visit of Lieutenant Governor Amy Tucks, State of Mississippi (far right)

Article Continued from Page 19

The QSUB Job (called news.script)

```
#QSUB -s /bin/csh
# Specifies the shell to
use

#QSUB -q batch
# Specifies the queue name

#QSUB -lT 330
# Specifies the per request
CPU time limit in seconds

#QSUB -lt 300
# Specifies the per process
CPU time limit in seconds

#QSUB -lM 100Mw
# Specifies the per request
memory limit inmegawords

#QSUB -lm 95Mw
# Specifies the per process
memory limit in megawords

#QSUB -o
/u/home/ray/news.job.output
# Directs stdout to the
stated file

#QSUB -eo
# Merges stderr and stdout
produced by the job

#

# Execute the script

#

/u/home/ray/news.host.csh
```

The Submission

```
athena%  qsub
~ray/news.script

nqs-181 qsub: INFO

  Request <16648.athena>:
Submitted to queue <nqenlb>
by <ray(297)>.

athena%
```

The Output (header has been deleted)

```
+ setenv HOST `hostname`
+ hostname
+ echo zeus
zeus
+ if ( zeus != athena )
then
+ if ( ! -d
/tmp/ray/round_error ) then
+ mkdir -p
/tmp/ray/round_error
+ endif
+ cd /tmp/ray/round_error
+ timex rcp athena-
hip0:/tmp/ray/round_error/in
put.f .
      seconds    "clocks"
real  5.928195   (592819510)
user  0.007392   (739175)
sys   0.065503   (6550348)
+ timex rcp athena-
hip0:/tmp/
ray/round_error/data.dat .
      seconds    "clocks"
real  5.132697   (513269738)
user  0.012876   (1287588)
sys   0.752989   (75298936)
+ else
+ pwd
/tmp/ray/round_error
+ f90 -o test.job input.f
+ chmod 755 test.job
+ ls -l
total 98784
-rw-r--r-- 1 ray root 49000000
Mar 27 15:10 data.dat
-rw-r--r-- 1 ray root 848 Mar
27 15:10 input.f
-rwxr-xr-x 1 ray root 1542376
Mar 27 15:11 test.job
+ ja
+ ./test.job
 forward sum is =
485778955946.541
 reverse sum is =
485778956355.709
+ ja -st
```

Job Accounting - Summary Report

```
Job Accounting File Name:/tmp/nqs.+++++2337/.jacct13117
Operating System:               unicos zeus 10.0.0.7 roo.4 CRAY SV1
User Name (ID):                 ray (297)
Group Name (ID):                usersup (139)
Account Name (ID):              NA0101 (50003)
Job Name (ID):                  news.script (13117)
Report Starts:                  03/27/01 15:11:04
Report Ends:                    03/27/01 15:11:42
Elapsed Time:                            38 Seconds
User CPU Time:                           36.7994 Seconds
System CPU Time:                         1.5433 Seconds
I/O Wait Time (Locked):                  0.0254 Seconds
I/O Wait Time (Unlocked):                0.0223 Seconds
CPU Time Memory Integral:                53.4371 Mword-seconds
SDS Time Memory Integral:                0.0000 Mword-seconds
I/O Wait Time Memory Integral:           0.0352 Mword-seconds
Data Transferred:                        5.8413 MWords
Maximum memory used:                     1.3945 MWords
Logical I/O Requests:                  1499
Physical I/O Requests:                      4
Number of Commands:                         2
Billing Units:                              0.0000
+ echo  End of Job
 End of Job
logout
athena%
```

Upcoming Events

August 2001

HPCD - 10th International Symposium on High-Performance Distributed Computing
7-10 August ❦ San Francisco, California
Ian Foster, itf@mcs.anl.gov

PDCS 2001 - 14th Annual International Conference on Parallel and Distributed Computing Systems
8-10 August ❦ Dallas, Texas
Edwin Sha, edsha@utdallas.edu

DS-RT 2001 - 5th IEEE International Workshop on Distributed Simulation and Real-Time Applications
13-15 August ❦ Cincinnati, Ohio
Mark Pullen, mpullen@gmu.edu
www.cs.unt.edu/~boukerch/DS-RT2001

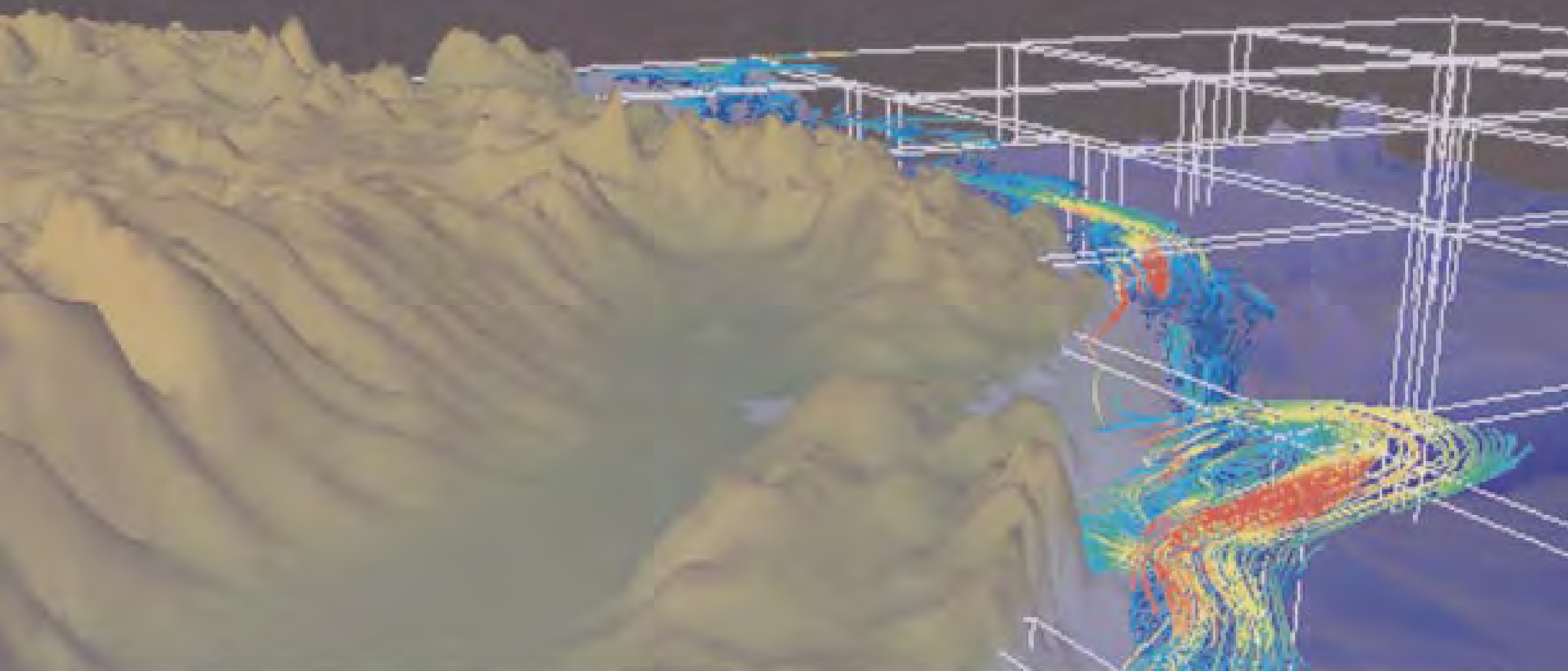
13th International Conference on Parallel and Distributed Computing and Systems
21-24 August ❦ Anaheim, California
Carrie Manchuck, calgary@iasted.com
www.iasted.com/conferences/2001/anaheim/pdcs.htm

September 2001

PARCO2001 - Conference on Parallel Computing
04-07 September ❦ Naples, Italy
www.parco.org

November 2001

Beyond Boundaries - Scientific Computing Conference
10-15 November 2001 ❦ Denver, Colorado
www.sc2001.org



Naval Oceanographic Office * MAJOR SHARED RESOURCE CENTER

1002 Balch Boulevard . Stennis Space Center, Mississippi . 39522